

**Portable PCMCIA interface for a host computer.**

Patent Number: ☐ EP0672987, A3

Publication date: 1995-09-20

Inventor(s): BENDER MICHAEL S (US); PATTON CHARLES F (US); MCCALLUM DOUGLAS (US); VO DUONG MINH (US)

Applicant(s): SUN MICROSYSTEMS INC (US)

Requested Patent: ☐ JP8036539

Application Number: EP19950301603 19950310

Priority Number (s): US19940213752 19940314

IPC Classification: G06F13/10

EC Classification: G06F13/10D


Equivalents: CA2143492, ☐ US5519851

Cited Documents:

---

**Abstract**

---

A portable PCMCIA interface for a host computer having a system bus. In one embodiment, the host computer is a SPARC based computer having an SBus and running the UNIX operating system. The PCMCIA interface provides a user application with access to a PCMCIA card. In this embodiment, the PCMCIA interface includes software and hardware components. The software component includes a hardware-independent portion and a hardware-dependent portion. By implementing the software in a suitable high level language such as "C", the software can be easily ported to other hardware platforms by merely adapting the hardware-dependent portion. The hardware component includes an ASIC coupled between the system bus and a couple of PCMCIA sockets. In some embodiments, the hardware also includes a 5 volt to 12 volt DC-DC converter between the system bus and the PCMCIA sockets. 

---

Data supplied from the esp@cenet database - I2

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-36539

(43)公開日 平成8年(1996)2月6日

(51)Int.Cl. <sup>6</sup>	識別記号	庁内整理番号	F 1	技術表示箇所
G 0 6 F 13/10	3 2 0 Z	7368-5E		
	3/00	G		
	3/08	C		
G 0 6 K 17/00	C			

審査請求 未請求 請求項の数 8 F D (全 20 頁)

(21)出願番号 特願平7-80873

(22)出願日 平成7年(1995)3月14日

(31)優先権主張番号 2 1 3 7 5 2

(32)優先日 1994年3月14日

(33)優先権主張国 米国 (U S)

(71)出願人 591064003

サン・マイクロシステムズ・インコーポレ  
ーテッド

SUN MICROSYSTEMS, IN  
CORPORATED

アメリカ合衆国 94043 カリフォルニア  
州・マウンテンビュー・ガルシア アヴェ  
ニュー・2550

(72)発明者 マイケル・エス・ベンダー

アメリカ合衆国 95006 カリフォルニア  
州・ボルダー クリーク・サンビーム ア  
ヴェニュー・155

(74)代理人 弁理士 山川 政樹

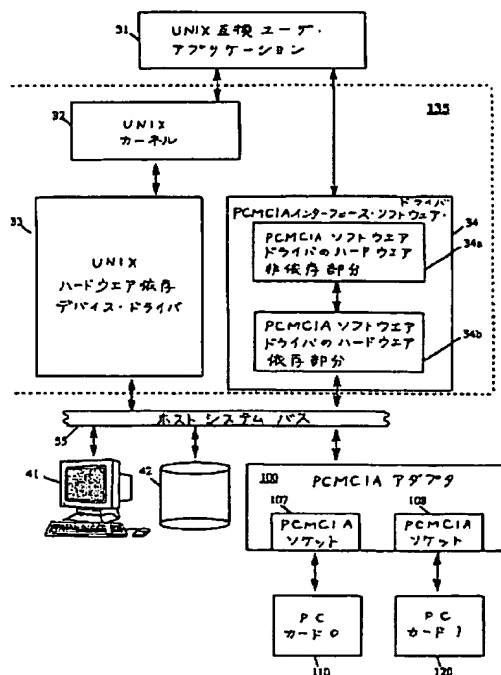
最終頁に続く

(54)【発明の名称】 PCMCIAインタフェース及びユーザ・アプリケーションとPCカードの間の通信を行う方法

#### (57)【要約】

【目的】 システム・バスを有するホスト・コンピュータ用の移植性の高いPCMCIAインタフェースを提供する。

【構成】 ホスト・コンピュータはシステム・バスを有しており、UNIXオペレーティング・システムを実行するSPARCベースのコンピュータである。PCMCIAインタフェースはユーザ・アプリケーションをPCMCIAカードにアクセスさせる。PCMCIAインタフェースはソフトウェアおよびハードウェアの構成要素を含んでいる。ソフトウェア構成要素は、ハードウェア非依存部分とハードウェア依存部分を含んでいる。ソフトウェアを「C」言語などの適切な高水準言語で実現することによって、ハードウェア依存部分を適合させるだけで、ソフトウェアを他のハードウェア・プラットフォームに簡単に移植できる。



## 【特許請求の範囲】

【請求項1】 ホスト・コンピュータで稼動しているユーザ・アプリケーションとPCカードの間の通信をもたらすPCMCIAインタフェースにおいて、前記ユーザ・アプリケーションからの外部PCMCIAアクセス要求を処理して、対応する内部PCMCIAアクセス要求を生成するハードウェア非依存ネクサスと、そのハードウェア非依存ネクサスに結合され、前記内部PCMCIAアクセス要求を処理し、対応するシステム・バス信号をシステム・バスに発生させるハードウェア依存ドライバと、前記システム・バスと前記PCカードの間に結合され、前記システム・バス信号を前記PCカードのためのPCカード信号に変換するPCMCIAアダプタとを備えていることを特徴とするPCMCIAインタフェース。

【請求項2】 ホスト・コンピュータで稼動しているユーザ・アプリケーションとそのホスト・コンピュータのシステム・バスに結合されたPCカードの間の通信を行う方法において、外部PCMCIAアクセス要求を処理し、その外部PCMCIAアクセス要求に応じて内部PCMCIAアクセス要求を生成し、その内部PCMCIAアクセス要求を処理し、SNO内部PCMCIAアクセス要求に応じて前記システム・バス上にシステム・バス信号を発生し、そのシステム・バス信号をそのシステム・バス信号に応じたPCカード信号に変換するコンピュータで実現されたステップを備えていることを特徴とする方法。

【請求項3】 前記ユーザ・アプリケーションに対する非送信請求レポートを生成するための通信チャネルを前記PCカードにもたらしめるためのコンピュータで実現されたステップをさらに含んでいる請求項2に記載の方法。

【請求項4】 前記PCカードからの情報タブルを解析するためのコンピュータで実現されたステップをさらに含んでいる請求項2に記載の方法。

【請求項5】 ホスト・コンピュータのシステム・バスとPCカードの間をインタフェースするPCMCIAアダプタにおいて、

前記PCカードに適合するためのPCMCIAソケットと、

前記システム・バスと前記PCMCIAカードの間で転送される情報を格納するためのバッファと、

前記情報の転送を制御するためのコア論理とを備えていることを特徴とするPCMCIAアダプタ。

【請求項6】 前記システム・バスと前記PCMCIAソケットの間への電力の印加および除去を行うスイッチをさらに含んでいる請求項5に記載のPCMCIAアダプタ。

【請求項7】 前記PCMCIAソケットに結合されたDC-DCコンバータをさらに含んでいる請求項5に記

載のPCMCIAアダプタ。

【請求項8】 前記PCカードから検索されたブート・イメージによって前記ホスト・コンピュータをブートするブートPROMをさらに含んでいる請求項5に記載のPCMCIAアダプタ。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明はホスト・コンピュータと周辺装置の間のインタフェースに関する。具体的にいえば、本発明はホスト・コンピュータとパーソナル・コンピュータ・メモリ・カード国際協会(PCMCIA: Personal Computer Memory Card International Association)の周辺カードとの間のインタフェースに関する。

【0002】

【従来の技術】 図1はスケーラブル・プロセッサ・アーキテクチャ(SPARC)ホスト・コンピュータ(SPARCはSPARC International, Inc.の登録商標である)などの周知の非x86ベースのホスト・コンピュータのハードウェア構成要素を示すブロック図である。ホスト・コンピュータ10はホスト中央演算処理装置(CPU)20、メモリ・バス(M-バス)35、ホスト・メモリ30、ホスト・バス・コントローラ50、標準ホスト・システム・バス55ならびにモニタ/キーボード/マウス41およびハード・ディスク装置42などの入出力(I/O)装置を含んでいる。システム・バス55はCPU20とI/O装置41、42を互いに接続し、これらの間の通信を行う。ホスト・コンピュータの中には、システム・バス55がSPARCベースのバスのものである。

【0003】 図2は図1のホスト・コンピュータのソフトウェア構成要素を示す図である。UNIXユーザ・アプリケーション31はカーネル32およびハードウェア依存ドライバ33を有するUNIXオペレーティング・システム(O/S)130と、システム・バス55に結合されている(UNIXはNovell, Inc.の100%子会社であるUNIX System Laboratories, Inc.の登録商標である)。システム・バス55はI/O装置41、42に結合されている。通常、ホスト・コンピュータ10が初めて初期化されると、UNIX O/S130がホスト・メモリ30にロードされ、コンピュータ10の電源が落とされるまで、常駐するようになる。したがって、I/O装置41、42に対応するI/Oデバイス・ドライバ33がロードされ、ユーザ・アプリケーション31がカーネル32に対して適切な呼び出しを行うことによってI/O装置41、42にアクセスする。カーネル32はアクセスをデバイス・ドライバ33、システム・バス55を介してI/O装置41、42に通信する。

【0004】 ホスト・コンピュータ10などのコンピュータの処理能力が高まったことに対応して、アプリケーション31などのユーザ・アプリケーションも複雑とな

ってきた。その結果、システム・バス55に結合される外部装置のもっとも一般的な形態である追加メモリ、大容量記憶装置および通信装置に対する必要性が高まってきた。通常、外部メモリや大容量記憶装置をホスト・コンピュータ10に追加するには、プリント回路板(PCB)の形態をしたアダプタを、コンピュータ10のハウジング内に配置されている利用可能なシステム・バス・スロットに挿入することを伴っている。システム・バス・スロットに挿入されると、PCBはシステム・バス55に電気的に結合され、アダプタはしたがってホストC  
PU20と通信を行えるようになる。それ故、ホスト・コンピュータ10に結合できる外部装置の最大数はコンピュータ10内で利用できるシステム・バス・スロットの総数によって制限される。さらに、典型的な技術的素養のないユーザはPCBの挿入や除去に関する訓練を受けておらず、訓練を受けている技術系のユーザがこのようなグレードアップや変更を行う必要がある。

【0005】一方、IBMパーソナル・コンピュータなどのIntel x86ファミリーのマイクロプロセッサをベースとした異なるハードウェア分野のコンピュータにおいて、外部装置を追加するための互換性および相互操作性の両方を促進するためにPCMCIA仕様で作成された(IntelはIntel Corporationの登録商標であり、IBMはInternational Business Machines Corporationの登録商標である)。典型的には、PCMCIAアダプタはPCMCIAカード(PCカード)を差し込むためのソケットを備えており、ホスト・コンピュータ上で作動する関連したインタフェース・ソフトウェアによって制御される。それ故、エンド・ユーザがPCカード様式で実現された広範囲に及ぶ各種の外部周辺装置およびメモリ装置を、必要に応じ、x86ベースのホスト・コンピュータに挿入したり、交換したりすることを簡単に行えるようになっている。

【0006】当初、x86ベースのPCMCIAインタフェースは比較的高速なランダム・アクセス・メモリ(RAM)カードを対象として設計されたものであり、それ故、アダプタとx86ベースのホスト・コンピュータとの間の指定された読み書きプロトコルはメモリ・タイプのアクセス操作に必要なものと同様なものであった。メモリを備えたPCカードにはRAM PCカードやハード・ディスク装置PCカードなどがある。最近、PCMCIA仕様は拡張され、FAXモデム・カードなどのI/Oタイプのデータ・アクセスを必要とするPCカードに適合するようになっている。

【0007】図3はISAバスやVESAバスなどのx86ベースのコンピュータのシステム・バス用に特に設計されたPCMCIAインタフェースを有するこのような従来技術のx86ベース・ホスト・コンピュータのソフトウェアおよびハードウェア両方の構成要素を示すブロック図である。ソフトウェア構成要素にはユーザ・ア

プリケーション141、Microsoftのディスク・オペレーティング・システム(DOS)、およびx86ベース・ハードウェア依存PCMCIAソフトウェア・ドライバ143などがある(MicrosoftはMicrosoft Corporationの登録商標である)。ハードウェア構成要素にはx86ベース・システム・バス145、一対のPCカード110、120を収納するための対応する対のPCMCIAソケット147a、147bを有するx86ベースPCMCIAアダプタ146などがある。x86ベースPCMCIAアダプタ用の従来技術の集積回路(IC)の2つの例はIntelの82365SL PCカード・インタフェース・コントローラとCirrus LogicのCL-PD6710/20である。

【0008】ユーザ・アプリケーション141はまず標準化されたシステム・コールをDOS142に対して行ってPCカード110または120にアクセスする。次に、DOS142は適切なPCMCIA呼び出しをx86ベースPCMCIAソフトウェア・ドライバ143に対して行って、アクセスをx86ベース・システム・バス145を介してPCMCIAアダプタ146に伝える。バス145からアクセス要求を受け取ると、アダプタ146はシステム・バス信号をPCカード110または120に対する標準PCMCIA信号に変換する。

【0009】

【発明が解決しようとする課題】従来のPCMCIAインタフェースに付随する問題の1つは、PCMCIA仕様がDOS、MS-WindowsまたはOS/2などのオペレーティング・システムを作動させるIntelのx86ファミリーのマイクロプロセッサをベースとしたコンピュータ向けに本来開発されたものであることである(MS-WindowsはMicrosoft Corporationの商標である)。したがって、PCMCIA仕様は全体的に、あるいは部分的にx86アセンブラ言語でコーディングされたソフトウェア・ドライバに合わせて最適化されており、これらのドライバは非x86ベースのコンピュータ・システム用に簡単に移植することはできない。

【0010】UNIXやDOSなどのもっとも一般的なオペレーティング・システムでの他の問題は、ホスト・コンピュータの作動中にハードウェア装置のランダムな接続除去を任意に行えるようにはこれらのオペレーティング・システムが設計されていないことである。従来は、システムのブート時に、各デバイス・ドライバをボーリングして、対応するハードウェア装置が存在しているかどうかを判定している。装置が除去されていたり、作動しない場合には、デバイス・ドライバはホスト・コンピュータのメモリからアンロードされるか、あるいはメモリに残されたとしても、使用されなくなるかする。この特に従来のオペレーティング・システムの特性によ

5

って、PCカードを安全に「ホット・プラグング」できなくなる。すなわち、ホスト・コンピュータの作動中にPCMCIAソケットに対するPCカードの挿入除去を行うことができない。

【0011】それ故、ソフトウェアおよびハードウェアのPCMCIA仕様を完全にサポートし、ホスト・コンピュータのバス・アーキテクチャ、プロセッサまたはオペレーティング・システムに無関係な移植性の高いPCMCIAインタフェースが必要である。このような移植性の高いPCMCIAインタフェースはSPARCなどのハードウェア・アーキテクチャやUNIXなどのオペレーティング・システムをベースとした他のホスト・コンピュータが、数が増加している標準PCカードを利用できるとともに、ホット・プラグングなどのPCMCIAの機能を完全にサポートするのを可能とするものである。

【0012】

【課題を解決するための手段】本発明はユーザ・アプリケーションをPCカードに結合するために、多くのホスト・コンピュータで作動できる移植性の高いPCMCIA

インタフェースを提供する。

【0013】本発明において、ユーザ・アプリケーションはホスト・システム・バスを有しているホスト・コンピュータで作動する。PCMCIAインタフェースはソフトウェアおよびハードウェア両方の構成要素を含んでいる。PCMCIAインタフェース・ソフトウェアはハードウェア非依存部分とハードウェア依存部分を含んでおり、ユーザ・アプリケーションとハードウェア非依存部分の間には定義済みの外部インタフェースがある。外部PCMCIAソフトウェア・インタフェースはPCMCIAカード・サービスおよびソケット・サービスを、基礎となるハードウェア、たとえばPCカードに透過的な態様で、ユーザ・アプリケーションに提供する。さらに、PCMCIAインタフェース・ソフトウェアのハードウェア非依存部分とハードウェア依存部分の間には定義済みの内部インタフェースがあり、ハードウェア依存部分を適合させるだけで、PCMCIAインタフェース全体を他のホスト・コンピュータに簡単に移植できるようにしている。

【0014】PCMCIAインタフェース・ハードウェアはシステム・バスに結合されたPCMCIAアダプタを含んでおり、アダプタはPCカードに適応する少なくとも1つのPCMCIAソケットを有している。ユーザ・アプリケーションはPCMCIA仕様によって指示されているとおりにPCカードとの間のすべてのデータの転送を開始する。ホスト・システム・バス互換PCMCIAアダプタはPCカードの属性メモリ・スペース、共通メモリ・スペースおよびI/Oスペースとの間の8ビット、16ビットおよび32ビットのデータ転送をサポートする。

6

【0015】実施例によっては、システム・バス電源とPCMCIAアダプタのPCMCIAソケットの間に電源スイッチ回路を追加することで、ホスト・コンピュータの電源を落とす必要なしに、PCカードをPCMCIAソケットに対して安全に挿入除去を行えるようになっている。オプションの5ボルトから12ボルトへのDC-DCコンバータが12ボルトの電源を必要とするPCカードに付加的な互換性をもたらす。

【0016】本発明のPCMCIAインタフェースは従来技術に比較して多くの利点を有している。PCMCIAインタフェース・ソフトウェアのハードウェア非依存部分とハードウェア依存部分との間に定義済みの内部インタフェースを設けることによって、システム・バス、アダプタおよびPCカードの個々の特性がホスト・コンピュータで作動するユーザ・アプリケーションにとって透過性のものとなる。さらに、高水準プログラミング言語でソフトウェアを実現することによって、ハードウェア依存部分を適合させるだけで、PCMCIAインタフェース・ソフトウェア全体を異なるオペレーティング・システムおよびハードウェア・アーキテクチャを有する他のホスト・コンピュータに移植することができる。したがって、本発明のPCMCIAインタフェースは他のホスト・コンピュータに簡単に移植できるハードウェアおよびソフトウェアのソリューションを提供する。

【0017】本発明のシステムの目的、特徴および利点は以下の説明より明らかとなろう。

【0018】

【実施例】図4には、本発明のPCMCIAアダプタに結合されたスケーラブル・プロセッサ・アーキテクチャ(SPARC)ベースのホスト・コンピュータが示されている。SPARCコンピュータ・アーキテクチャの詳細については、SPARC International, Menlo Park, Californiaから入手可能なSPARC Architecture Manual, Version 8, 1992を参照されたい。SPARCベースのホスト・コンピュータ10の作動は周知であるから、本明細書では詳細な説明を行わない。

【0019】本発明の1実施例によれば、ホスト・コンピュータ10はシステム・バス55を介してPCMCIAアダプタ100に結合されている。アダプタ100はPCカードの対110、120を収納するための対応するPCMCIAソケットの対107、108を有している。PCカード110、120はPCMCIAソケット107、108内に配置された対応する対となっている68ピン・コネクタを介してアダプタ100に挿入される。

【0020】アダプタ100が各種の物理的構成をとりうることに留意すべきである。たとえば、デスクトップ・コンピュータ用の1実施例において、アダプタ100は周辺プリント回路板(PCB)に実装された特定用途向け集積回路(ASIC)を含んでおり、PCBはシス

テム・バス55内の物理的スロットを占有している。電気ケーブルがASICをPCMCIAソケット107、108に接続する。バームトップ・コンピュータ用の他の実施例において、アダプタ100はCPU20とともにメインPCB上におかれたASICを含んでいる。電気ケーブルがASICを、バームトップ・コンピュータのハウジングに取り付けられたPCMCIAソケット107、108に接続する。あるいは、PCMCIAソケット107、108をASICおよびCPU20とともに、デスクトップまたはラップトップ・コンピュータのメインPCBに取り付け、これによって、相互接続電気ケーブルを不必要とする。他の実施例および改変形も可能であり、これらは本明細書の開示から当分野の技術者に明らかとなろう。

【0021】ホスト・コンピュータ10で稼動し、PCMCIAアダプタ100を制御するPCMCIAインタフェース・ソフトウェア・ドライバの作動を説明するために、主としてPCカード110およびこれに関連するハードウェアおよびソフトウェア・ドライバを参照する。PCカード110、120の両方がそれぞれPCMCIAソケット107、108に電気的および物理的に結合されているので、第1のPCカード110に関するソフトウェア・ドライバおよびアダプタ100の説明は同じように第2のPCカード120に適用できる。

【0022】本実施例において、図5のブロック図に示すように、UNIXカーネル32およびデバイス・ドライバ33を含んでいるUNIXオペレーティング・システム135は、PCMCIAインタフェース・ソフトウェア・ドライバ34も含んでいる。UNIX互換ユーザ・アプリケーション31はPCMCIAインタフェース・ソフトウェア・ドライバ34、システム・バス55およびアダプタ100を介してPCカード100にアクセスする。ソフトウェア・ドライバ34はハードウェア非依存部分34aおよびハードウェア依存部分34bを含んでいる。

【0023】インタフェース・ソフトウェア・ドライバ34はC言語などの適当な高水準プログラミング言語でコーディングされており、ソース・コードをハードウェア依存部分34bに合わせて変更し、ドライバ34を再コンパイルするだけで、ドライバ34のソース・コードを簡単に他のホスト・コンピュータ・プラットフォームに移植することが可能となっている。他に考えられるホスト・コンピュータの例としては、x86マイクロプロセッサ・ベースのコンピュータのUNIXオペレーティング・システムがある。ホスト・コンピュータ・プラットフォームの他の変形および改変形は、当分野の技術者にとって明らかなものであろう。

【0024】図6はPCMCIAインタフェース・ソフトウェア・ドライバ34の詳細を示すブロック図である。ハードウェア非依存部分34aは一对のPCカード

・ドライバ260、270、PCMCIAネクサス・ドライバ210、イベント・マネージャ230およびカード・サービス層220を含んでいる。ハードウェア依存部分34bは、アダプタ100を制御するためのシステム・バス互換PCMCIAアダプタ・ドライバ250を備えている。ハードウェア非依存ネクサス・ドライバ210とハードウェア依存システム・バス互換アダプタ・ドライバ250の間の内部インタフェース250を、本発明の1態様にしたがって明確に定義することにより、インタフェース・ソフトウェア・ドライバ34のハードウェア依存部分34bを適合させるだけで、PCMCIAインタフェース全体を異なるホスト・コンピュータ・プラットフォームに有利に移植することができる。

【0025】ユーザ・アプリケーション31がPCカード110にアクセスを開始する前に、PCMCIAインタフェース・ソフトウェア・ドライバ34をまずホスト・メモリ30にロードしなければならない。ソフトウェア・ドライバ34の各種の部分のローディングは、ホスト・コンピュータ10の初期化中に達成される。PCMCIAネクサス・ドライバ210がまずロードされると、ネクサス・ドライバ210は構成ファイルに格納されている使用可能なすべてのアダプタ・ドライバのリストを探索し、各アダプタ・ドライバを構成ファイルにロードしようと試みる。特定のアダプタ・ドライバ、たとえば、アダプタ・ドライバ250が正常にロードされるたびに、ネクサス・ドライバ210は各ドライバ、たとえば、アダプタ・ドライバ250に関連する対応したドライバ装置構造体、たとえば構造体251を探し出す。アダプタ・ドライバ250は次いで、対応するドライバ装置構造体251に対するポインタを保存する。

【0026】次に、PCMCIAネクサス・ドライバ210はPCMCIAアダプタ・ドライバ250を照会して、その基本アダプタ特性を取得し、PCMCIAソケット107、108を利用可能な論理ソケットのリストに追加する。ネクサス・ドライバ210は次いで、論理ソケット・リストをカード・サービス層220へエクスポートする。したがって、カード・サービス層220はネクサス・ドライバ210とアダプタ・ドライバ250の間におかれているハードウェア非依存ソフトウェア・インタフェース215にアクセスし、カード・サービス層220がハードウェアに依存しない態様でアダプタ100およびPCMCIAソケット107のリソースを制御し、管理することを可能とする。ネクサス・ドライバ210は同じ論理ソケット・リストをアダプタ・ドライバ250にもエクスポートし、カード・サービス層220とアダプタ・ドライバ250の間に1対1の論理ソケット通信チャネルを確立する。

【0027】カード・サービス層220はカード情報構造体(CIS)インタプリタ220aも含んでおり、このインタプリタはPCカード110に自動識別情報をそ

のCISに維持するよう要求することによって、PCカード110がホスト・コンピュータのアーキテクチャやオペレーティング・システムに関係なく自動識別を行うことを可能とする。CISはPCカード110の属性メモリ・スペースに格納され、タブルという可変長要素の単一リンク・リストで構成されている。インタプリタ220aはタブル・パーサであり、すべてのタブル情報の処理を担当する。したがって、PCカード・ドライバ260はタブル構文解析コードを必要としない。

【0028】本実施例において、各タブルは長さが1バイトであり、最大256個のタブルがPCカード110のCISに格納される。タブルがインタプリタ220aによって構文解析され、認識されると、インタプリタ220aはPCカード110からのタブル・データをインタプリタ・リンク・リスト220bのタブル項目にコピーさせる。逆に、インタプリタ220aがタブルを認識しない場合には、タブルが認識されないこと、およびPCカード110からのタブル・データをインタプリタ・リンク・リスト220bのタブル項目にコピーすべきではないことを示すフラグがセットされる。PCカード110のCISの構文解析が正常に行われると、リンク・リスト220bはPCカード110の構成パラメータ、たとえば、PCカードのタイプ、メモリ容量およびアクセス速度などを含む。その後、カード・サービス層220はリンク・リスト220bを使用して、PCカード・ドライバ260からのカード・サービス要求を処理する。

【0029】実施例によっては、PCMCIAネクサス・ドライバ210がアダプタ・ドライバ250とその私用インタフェース215をPCカード・ドライバ260にエクスポートしないものもある。カード・サービス層220とPCMCIAネクサス・ドライバ210の間のインタフェース225も私用である。たとえば、PCカード・ドライバ260はPCMCIAネクサス・ドライバ210に対する直接呼出しを行わない。その代わり、ユーザ・アプリケーション31がPCカード110にアクセスするたびに、PCMCIAネクサス・ドライバ210宛のPCカード・ドライバ260に対するすべての呼出しが、カード・サービス層220を介して行われることになる。したがって、カード・サービス層220はPCカード・ドライバ260に、要求された機能に基づく可変引数リストとの単一の入口点をもたらす。

【0030】次に、PCMCIAが指定したイベント・コールバック機能をサポートするために、イベント・マネージャ230が個別のSTREAMSとしてロードされ、ネクサス・ドライバ210がPCカードの挿入/除去(ホット・プラグング)などのPCMCIAイベントをユーザ・アプリケーション31に伝えることを可能とする。(STREAMSはプロセスとデバイス・ドライバの間のUNIXの全二重接続である。)それ故、ネク

サス・ドライバ210は効率のよいイベント・コールバック機構をユーザ・アプリケーション31に提供し、イベント・マネージャ230はイベントを監視し、アダプタ・ドライバ250によってアダプタ100とPCカード110の両方を管理する。利用可能なPCカードの数によって、ユーザ・アプリケーション31がすべての可能なPCMCIAカードの対するポーリングを行うことを非実効的で、効率の悪いものとするため、イベント・マネージャ230は効率のよい解決策である。さらに重要なのは、イベント・マネージャ230がユーザ・アプリケーション31への非送信請求フィードバックのためのPCMCIA指定のチャネルを実現するための単一の機構(通常、UNIXではサポートされていない)をもたらすことである。

【0031】たとえば、PCカード110のPCMCIAソケット107への挿入が成功した場合、ネクサス・ドライバ210はアダプタ・ドライバ250から「カード挿入」イベント通知を受け取り、ネクサス・ドライバ210がどのタイプのPCカードがソケット107に入っているのかを追跡できるようにする。その後、対応するPCカード・ドライバ、たとえば、ドライバ260がロードされ、ネクサス・ドライバ210が対応するドライバ装置構造体、たとえば構造体251を更新し、これによってPCMCIAソケット107とPCカード110の間のアソシエーションを形成する。このイベント・コールバック機能をサポートするためのPCMCIAインタフェース・ハードウェアについての検討は、アダプタ100のハードウェアについての以下の説明で行う。

【0032】PCMCIAインタフェース・ソフトウェア・ドライバ34のローディングおよび初期化が完了すると、ユーザ・アプリケーション31はアダプタ100およびPCカード110にアクセスする。図7は本発明によるホスト・コンピュータ10で稼動しているユーザ・アプリケーション31によるPCカード110のアクセスを説明するための流れ図である。

【0033】まず、ユーザ・アプリケーション31はPCカード・ドライバ260に対して外部ハードウェア非依存PCMCIA呼出しを行う。カード・ドライバ260は適切なカード・サービスをカード・サービス層220に要求することによって応答する。カード・サービス層220はPCカード・ドライバ260からのすべてのカード・サービス要求を処理し、適切な呼出しをPCMCIAネクサス・ドライバ210に対して行う。PCカード・ドライバ260はカード・サービス層220にPCカード・ドライバのデバイス情報ポインタ(DIP)に対するポインタを与え、これによってPCカード110を一意に識別し、PCカード・ドライバ260の親プロセス、すなわちネクサス・ドライバ210に対する経路を与える。カード・サービス層220は次いでDIPを使用して、ネクサス・ドライバ210を探し、これに

11

対して適切な呼出しを行う。このような呼出しには、アダプタ・ドライバ250によって提供されるソケット・サービス宛のアダプタ・ハードウェア構成要求がある。

【0034】カード・サービス層220からの適切な呼出しに回答して、ネクサス・ドライバ210はPCMCIAアダプタ・ドライバ250に対する対応した内部ハードウェア非依存ソフトウェア呼出しを生成する。ハードウェア依存アダプタ・ドライバ250は、次いで、内部ソフトウェア呼出しをアダプタ100用の適切なシステム・バス信号に変換する。アダプタ100はその後、PCMCIAソケット107にあるPCカード110に対する適切なPCMCIAカード信号を生成する。

【0035】要約すると、ユーザ・アプリケーション31はハードウェア非依存外部PCMCIA呼出しの形態の要求を、PCMCIAインタフェース・ソフトウェア・ドライバ34に伝えることによって、PCカード110とのデータ・アクセスを開始する。次いで、ホスト・メモリ30に常駐しているソフトウェア・ドライバ34が外部PCMCIA呼出しを受け取り、適切な依存アダプタ固有信号をシステム・バス55でアサートさせる。ユーザ・アプリケーション31からのアクセス要求を適切なシステム・バス信号に変換する際のインタフェース・ソフトウェア・ドライバ34の各種の部分がもたらす作動およびサービスを説明してきたが、以下ではPCMCIAインタフェース・ハードウェア、すなわちアダプタ100およびPCカード110の作動を詳細に説明する。

【0036】図8に示すような実施例において、PCMCIAアダプタ100はホスト・バス・インタフェース・バッファ101、アダプタ・コア論理102、およびカード・インタフェース・バッファ103を有する特定用途向けIC(ASIC)100aを含んでいる。アダプタ100は電源スイッチ105、PROM106、および一対のPCMCIAソケット107、108も含んでいる。PROM106はオープン・ブートPROM(OBP)で、標準的なブートROM機能をホスト・コンピュータ10に与える。

【0037】図9はPCカード110、120の各々に割り振られた、すなわち、6メガバイトのアドレス・スペースがPCカード110、120専用となっているホスト・メモリ30の3メガバイトのシステム・バス・スペースを示すアドレス・マップである。

【0038】本実施例において、メモリ30のアドレス・スペース0ないしFFFFFFhおよび3FFFFFFhないし4FFFFFFhはそれぞれPROM106ならびに制御および状況レジスタ(CSR)102aの内容へのアクセス用に予約されている。PCMCIA仕様は最大各64メガバイトの属性メモリ・スペース、共通メモリ・スペースおよびI/Oメモリ・スペースもサポートしている。しかしながら、PCカード110の前述の実

12

施例においては、各1メガバイトのシステム・バス・アドレス・スペースだけが属性メモリ・スペース、共通メモリ・スペースおよびI/Oメモリ・スペースに予約されている。したがって、システム・バス・アドレス・スペースを指定されたPCMCIAアドレス・スペースに転置するためのアドレス・マッピングが必要となる。アダプタ100の8メガバイトのアドレス・スペースに対するアドレス・マップは、PROM106に格納されている。

【0039】CSR102aはアダプタ100を各種のタイプのPCカードとともに作動するように構成するパラメータを格納するために使用される。CSR102aは64メガバイトのPCカード・アドレス・スペースの各々に各1メガバイトのシステム・バス・アドレス・スペースをマッピングするのに必要なアドレス・オフセット・テーブルも収納している。これらのアドレス・オフセット・テーブル値は特定のPCカード、たとえば、FAXモデム・カード、RAMメモリ・カード、あるいはウィンチェスタ・ディスク装置カードに合うようにホスト・コンピュータ10によって変更できるものである。CSR102aに格納されている構成パラメータにはPCカードのデータ・アクセス速度があり、この速度はカードの機能によって比較的低速なモデム・カードから比較的高速なRAMカードまでの範囲で変化できるものである。

【0040】図10はユーザ・アプリケーション31によるPCカードの読み取りおよび書き込みアクセスに応じた、システム・バス55における従来のPCMCIAの読取りおよび書き込みアクセス・サイクルを説明するための2つのタイミング図を示す。データ転送がPCMCIA仕様バージョン2.1で規定されているようにユーザ・アプリケーション31によって開始されることに留意されたい。たとえば、PCカード110の読取りアクセスに応じて、アダプタ100はPCMCIAソケット107のカード・アドレス・ラインADDRとカード・イネーブル・ラインCE' ( 'は信号が活動時に低であることを示す。以下同じ) をアサートする。アダプタ100は次いで、カード・アドレス・ラインADDRによって示される位置におけるPCカード110の読取りアクセスを示す出力イネーブル信号OE' をアサートする。特定の時間(すなわち、PCカードの読取りアクセス時間)の後、PCカード110は要求されたデータをカード・データ・ラインDATAへ送る。同様に、PCカード110の書き込みアクセスに応じて、アダプタ100はカード・データ・ラインDATA、カード・アドレス・ラインADDRおよびカード・イネーブル・ラインCE' をアサートし、その後、カード書き込みイネーブル・ラインWEがアサートされる。

【0041】本発明の1実施例によれば、PCカード110のバイト、ハーフワード(16ビット)およびワー



ド(32ビット)のデータ・アクセスをサポートする。PCMCIAソケット107がサポートできるのが16ビットのデータ転送だけであるから、サイズ調整はPCカード110の外部で行われる。アダプタ100の前述の実施例はいかなるデータ・バッキングも行わない(すなわち、単一のシステム・バス読取り要求に応じて、アダプタ100はカード110に対する複数の読取りアクセスを発生しない)が、当分野の技術者であれば、ホスト・コンピュータ10によるサイズ調整を必要とせず、16ビットよりも大きいワード長のデータ・アクセスをサポートしているアダプタ100の他の実施例に、作動の原理を拡張することができよう。

【0042】さらに、カード110の属性メモリ・スペースまたはI/Oメモリ・スペースで、アクセスが開始された場合、カード・メモリ選択ラインC0\_REG\*が低にアサートされる。逆に、アクセスが共通メモリ・スペースで開始された場合、カード・メモリ選択ラインC0\_REG\*が高であるとアサートされる。

【0043】図11のタイミング図を参照すると、アダプタ100を介してPCカード110の属性メモリ・スペースまたは共通メモリ・スペースのコンピュータ10によって開始されたバイト幅の読取りが示されている。まず、制御および状況レジスタ102aの対応するウィンドウ制御レジスタに格納されている基本アドレス値BASE、およびシステム・バス・アドレス・ラインSB\_PAのオフセット・アドレス値がカード・アドレス・ラインC0\_Aに出され、これによってシステム・バス・アドレスがPCカード・アドレスにマップされる。次に、カード出力イネーブル・ラインC0\_CE\*が低にアサートされる。L+1クロック・サイクル(Lはウィンドウ制御レジスタに格納されているCommand\_Strobe\_Delay値CMDLDLYである)後、カード出力イネーブル・ラインC0\_CE\* (図11ではC0\_RDCMD\*で示されている)が低にアサートされる。選択されたPCカード、すなわちPCカード110は次いで要求されたメモリ・データ・バイトを内部で検索し、データ値をカード・データ・ラインC0\_Dに出す。

【0044】他のM+1クロック・サイクル(ただし、Mはウィンドウ制御レジスタに格納されているCommand\_Strobe\_Length値CMDLNGである)の後、カード・データ・ラインC0\_D上のデータ・バイト値がアダプタ100によってラッチされ、カード出力イネーブル・ラインC0\_RDCMD\*はアサート解除される。1クロック後、カード出力イネーブル・ラインC0\_CE\*がアサート解除される。最後に、アダプタ100はシステム・バス・ラインSB\_ACK\*のバイト「ACK」をホスト・コンピュータ10に対してアサートし、ラッチされているデータ・バイトをさらに1クロック・サイクル後にシステム・バス・データ・ラインSB\_DATAに送り、これによってバイト幅の読取りサイクルを完

了する。

【0045】PCカード110のハーフワード幅の属性または共通メモリの読取りは次の通りである。まず、ウィンドウ制御レジスタのBASE値がカード・アドレス・ラインC0\_Aに印加され、システム・バス・アドレス・ラインSB\_PAのオフセット・アドレス値がカード・アドレス・ラインC0\_Aに印加され、カード・イネーブル・ラインC0\_CE\*が低にアサートされる。L+1クロック・サイクル後、カード出力イネーブル・ラインC0\_CE\* (図11では、C0\_RDCMD\*で示されている)が低にアサートされる。選択されたPCカード110が次いで、対応するメモリ・データ・バイトをカード・データ・ラインC0\_Dに出す。M+1クロック・サイクル後、カード・データ・ラインC0\_Dの有効なデータ・バイトがアダプタ100にラッチされ、カード・イネーブル・ラインC0\_CE\* (C0\_RDCMD\*参照)が次いでアサート解除される。1クロック・サイクル後、カード・イネーブル・ラインC0\_CE\*がアサート解除される。アダプタ100はシステム・バス・ラインSB\_ACK\*に対してハーフワードのACKをアサートし、ラッチされているデータ・バイトをシステム・バス・データ・ラインSB\_DATAとバイト・スワップさせ、これによってハーフワードの読取りサイクルを完了する。

【0046】PCカード110のワード幅の属性または共通メモリ・スペースの読取りは、ホスト・コンピュータ10に対してハーフワードのACKをアサートするところまで、ハーフワードの読取りアクセスと同様である。システム・バス55のハーフワードACKはホスト・コンピュータ10に、ワード読取りにはサイズ調整が必要なことを知らせる。したがって、コンピュータ10はシステム・バス55を介して他のアクセスを開始して、データの希望するワードの後半のハーフワードを読みとる必要がある。第2のハーフワードの第2の読取りサイクルは同じ態様で行われるが、システム・バス・アドレス・ラインSB\_PA[1]の値ならびにカード・アドレス・ラインC0\_A[1]の値が両方とも「0」ではなく「1」になっており、したがって、データの適切な次のハーフワードが取り出される点が異なっている。

【0047】前述したように、PCMCIA仕様は本来、外部RAMメモリ・カード用に開発されたものである。しかしながら、他の低速な記憶媒体、たとえば、ハード・ディスク装置が物理的に小さくなり、PCMCIAの形状係数に合うようになったため、PCMCIA仕様を拡張し、他の作動モードをサポートする、すなわち、「WAIT」機能を使用してPCカード110にアクセスすることが必要となった。本実施例においては、WAIT機能により、アダプタ100に図12に示すようにカード待機信号C0\_WAIT\*をアサートするこ

15

とによって、低速なPCカード110がアクセスに応答するのに遅くなってもよいようにさせることができる。

【0048】ホスト・コンピュータ10がPCカード110に対して、カード待機ラインC0\_WAIT<sup>\*</sup>を利用する読取りアクセスを開始した場合(wait\_requestビットWAITREQはウィンドウ制御レジスタで使用可能とされる)、システム・バス・アドレス・ラインSB\_PAのBASE値およびオフセット・アドレス値がカード・アドレス・ラインC0\_Aに印加され、カード・イネーブル・ラインC0\_CE<sup>\*</sup>は低にアサートされる。L+1クロック・サイクル後、カード出力イネーブル・ラインC0\_OE<sup>\*</sup>(図12では、C0\_RDCMD<sup>\*</sup>で示されている)が低にアサートされる。

【0049】M+N+2クロック・サイクル(Nはウィンドウ制御レジスタのWait\_Delay値WAITDLYに等しい)後、アダプタ100はカード待機ラインC0\_WAIT<sup>\*</sup>をサンプルする。カード待機ラインC0\_WAIT<sup>\*</sup>がアサートされている場合、PCカード110はメモリ・アクセスを完了するのに遅延を必要とする。PCMCIA仕様によれば、カード待機ラインC0\_WAIT<sup>\*</sup>を最長12マイクロ秒の間アサートしておくことができる。カード待機ラインC0\_WAIT<sup>\*</sup>の立ち上がり角がホスト・コンピュータ10のシステム・クロックに関して非同期であるから、PCカード110がカード待機ラインC0\_WAIT<sup>\*</sup>をアサート解除すると、アダプタ100は入来カード待機ラインC0\_WAIT<sup>\*</sup>を同期させなければならない。

【0050】次に、PCカード110は要求されたメモリ・データ・バイトをカード・データラインC0\_Dに対してアサートしてから、カード待機ラインC0\_WAIT<sup>\*</sup>をアサート解除する。アダプタ100はデータをラッチし、カード出力イネーブル・ラインC0\_OE<sup>\*</sup>(図10のC0\_RDCMD<sup>\*</sup>参照)をアサート解除する。1クロック・サイクル後、カード・イネーブル・ラインC0\_CE<sup>\*</sup>がアサート解除される。最後に、アダプタ100はシステム・バス・ラインSB\_ACK<sup>\*</sup>に「ACK」をアサートし、1クロック・サイクル後に、ラッチされたデータをシステム・バス・データ・ラインSB\_DATAに出す。

【0051】PCカード110の非待機バイトI/O読取りアクセスは上述の属性または共通メモリのバイト読取りのものと同様であるが、カード出力イネーブル・ラインC0\_OE<sup>\*</sup>ではなく、カードI/Oレディ・ラインC0\_IORD<sup>\*</sup>(図10では、C0\_RDCMD<sup>\*</sup>で示されている)が低にアサートされる点が異なっている。同様に、ハーフワードI/O読取りアクセスは属性または共通メモリのハーフワード読取りの場合と同様に開始されるが、カード・ラインC0\_IORD<sup>\*</sup>(C0\_RDCMD<sup>\*</sup>参照)がカード・イネーブル・ラインC0\_OE<sup>\*</sup>の代わりに使用される点が異なっている。

16

【0052】PCカード110が16ビットI/Oアクセスを行える場合、カード110は読取りアクセス中にカードI/O\_is\_16\_bitラインIOIS16(図示せず)を低にアサートし、データの両方のバイトがカード・データ・ラインC0\_Dに出される。すなわち、すべての16ビット・データ・ラインが有効となる。アダプタ100は次いで、システム・バス・ラインSB\_ACK<sup>\*</sup>を介してホスト・コンピュータ10にハーフワードACKを出し、1クロック・サイクル後に、ラッチされたデータをバイト・スワップして、システム・バス・データ・ラインSB\_DATAに出す。

【0053】逆に、PCカード110に行えるのがバイト・サイズのI/O読取りアクセスだけである場合には、カード110は読取りアクセス中にカード・ラインIOIS16<sup>\*</sup>を低にアサートせず、カード・データ・ラインC0\_Dの応答している8ビットに出されるデータ・バイトだけが有効となる。次に、アダプタ100はシステム・バス・ラインSB\_ACK<sup>\*</sup>を介してホスト・コンピュータ10にバイトACKをアサートし、これによってハーフワードI/Oアクセスにサイズ調整が必要なことを示す。ホスト・コンピュータ10は次いで、システム・バス55で他のアクセスを開始し、ハーフワードI/Oアクセスを完了するために第2のバイトを読みとる。

【0054】WAIT機能を使用したPCカード110のI/O読取りも、WAIT機能を使用した属性または共通メモリの読取りアクセスと同様であるが、カード・ラインC0\_OE<sup>\*</sup>ではなく、カードI/Oレディ・ラインC0\_IORD<sup>\*</sup>(図10では、C0\_RDCMD<sup>\*</sup>で示されている)が低にアサートされる点が異なっている。PCMCIA仕様は読取りアクセスの開始と終了の間に最長12マイクロ秒の遅延を認めている。しかしながら、カード・ラインC0\_WAIT<sup>\*</sup>を使用して読取りアクセスの完了を12マイクロ秒以上に遅らせることができるが、アダプタ100がシステム・バス55において分割読取りをサポートしている必要がある。読取りがシステム・バス55でホスト・コンピュータ10によって開始されると、カード待機ラインC0\_WAIT<sup>\*</sup>がPCカード110によってアサートされるまでにタイムアウトが発生した場合、再実行ACKがアダプタ100によってアサートされる。システム・バス55におけるホスト・コンピュータ10による以降の読取り再試行中に、分割読取りアクセスは完了する。次に、PCカード110はカード待機ラインC0\_WAIT<sup>\*</sup>をアサート解除し、読取りデータがシステム・バス・データ・ラインSB\_DATAに出される。

【0055】逆に、PCカード110がカード待機ラインC0\_WAIT<sup>\*</sup>をアサートしたが、読取りアクセス・サイクル中に20マイクロ秒以内にアサート解除しなかった場合、アダプタ100は読取りアクセス・サイク

ルを終了し、システム・バス 55 に割込みを行う。この割込みはホスト・コンピュータ 10 に WAIT タイムアウトが発生したことを通知する。

【0056】本発明のこの実施例によれば、図 13 のタイミング図で示すように、アダプタ 100 は PC カード 110 に対する非 WAIT バイト、ハーフワード、およびワード・データ書込みアクセスもサポートする。アダプタ 100 はシステム・バス 55 との、ハーフワード ACK を使用したワード・サイズ書込みアクセスに対してホスト・コンピュータ 10 によって外部サイズ調整の行われる 16 ビット幅のデータ・インタフェースを有している。上述したように、コンピュータ 10 が属性または共通メモリ、あるいは I/O アドレス・スペースで書込みアクセスを開始した場合、カード選択ライン C0\_R  
EG\* が低にアサートされる。

【0057】PC カード 110 の属性または共通メモリのバイト書込みがコンピュータ 10 によって開始された場合、システム・バス・アドレス・ライン SB\_PA からの BASE アドレス値およびオフセット・アドレス値が、カード・アドレス・ライン C0\_A に出され、システム・バス・データ・ライン SB\_DATA のデータ値がカード・データ・ライン C0\_D に送られ、カード出力イネーブル・ライン C0\_CE\* が低にアサートされる。次に、アダプタ 100 はバイト ACK をアサートし、これによってシステム・バス 55 における書込みサイクルを終了する。L+1 クロック・サイクル後、カード書込みイネーブル・ライン C0\_WE\* (図 11 では、C0\_WRCMD\* で示す) が低にアサートされる。M+1 クロック・サイクル後、カード書込みイネーブル・ライン C0\_WE\* が低であることからアサート解除される。P+1 クロック・サイクル (P は Recovery\_delay 値 RECDLY に等しい) 後、カード出力イネーブル・ライン C0\_CE\* がアサート解除される。

【0058】属性または共通メモリのハーフワード (16 ビット) 幅の書込みが開始されると、システム・バス・アドレス・ライン SB\_PA の BASE アドレス値とオフセット値がカード・アドレス・ライン C0\_A に出され、システム・バス・データ・ライン SB\_DATA のデータ値がバイト・スワップされてから、カード・データ・ライン C0\_D に出され、カード・イネーブル・ライン C0\_CE\* が低にアサートされる。次に、アダプタ 100 はシステム・バス・ライン SB\_ACK\* のハーフワード ACK をアサートし、これによってシステム・バス 55 における書込みサイクルを終了する。L+1 クロック・サイクル後、カード書込みイネーブル・ライン C0\_WE\* (図 13 では、C0\_WRCMD\* で示す) が低にアサートされる。M+1 クロック・サイクル後、カード書込みイネーブル・ライン C0\_WE\* がアサート解除される。最後に、P+1 クロック・サイクル後、カード・イネーブル・ライン C0\_CE\* がアサ

ート解除される。

【0059】属性/メモリのワード・サイズの書込みを開始するのに必要なイベントの順序は、PC カード 110 のハーフワード書込みのものと同様である。相違はアダプタ 100 がシステム・バス書込みサイクルを終了するハーフワード ACK をアサートし、これによってホスト・コンピュータ 10 にワード・サイズの書込みアクセスを完了するにはサイズ調整が必要であると通知することである。コンピュータ 10 はシステム・バス 55 によってデータの第 2 のハーフワードの書込みを開始することによって応答する。

【0060】図 12 のタイミング図を参照すると、アダプタ 100 は WAIT 機能を使用した PC カード 110 の属性または共通メモリに対するデータ書込みアクセスもサポートしている。カード待機ライン C0\_WAIT\* を利用する書込みアクセスが PC カード 110 で開始されると、システム・バス・ライン SB\_PA の BASE アドレス値およびオフセット・アドレス値がカード・アドレス・ライン C0\_A に出され、システム・バス・データ・ライン SB\_DATA のデータ値がカード・データ・ライン C0\_D に送られ、カード・イネーブル信号 C0\_CE\* が低にアサートされる。アダプタ 100 は次いで、適切な ACK、すなわちバイト書込みのバイト ACK およびハーフワードまたはワード書込みのハーフワード ACK を送り、これによってシステム・バス書込みサイクルを終了する。

【0061】L+1 クロック・サイクル後、カード・イネーブル・ライン C0\_WE\* (図 12 では、C0\_WRCMD\* で示す) が低にアサートされる。M+N+2 クロック・サイクル後、カード待機ライン C0\_WAIT\* がサンプルされる。カード待機ライン C0\_WAIT\* が「0」であれば、カード 110 はメモリ・アクセスの完了時の遅延を要求している。上述したように、PCMCIA 仕様では 12 マイクロ秒という最長待機期間が認められている。PC カード 110 がカード待機ライン C0\_WAIT\* をアサート解除すると、アダプタ 100 がカード書込みイネーブル・ライン C0\_WE\* (C0\_WRCMD\* 参照) をアサート解除する。P+1 クロック・サイクル後、カード・イネーブル・ライン C0\_CE\* はアサート解除される。

【0062】本実施例において、アダプタ 100 は図 15 のタイミング図に示すように、バイト・サイズ調整による PC カード 110 の非 WAIT I/O 書込みアクセスもサポートしている。PC カード 110 のバイト I/O 書込みアクセスは属性または共通メモリのバイト書込みと同様であるが、カード書込みイネーブル・ライン C0\_WE\* の代わりにカード I/O 書込みライン C0\_IOWR\* (図 15 では、C0\_WRCMD\* で示す) がアサートされる点が異なっている。同様に、PC カード 110 のハーフ I/O 書込みアクセスは属性また

19

は共通メモリのハーフワード書き込みと同様に開始されるが、カード書き込みイネーブル・ラインC0\_\_WE\*の代わりにカードI/O書き込みラインC0\_\_IOWR\* (C0\_\_WRCMD\* 参照) がアサートされる点が異なっている。

【0063】 上述したように、PCMCIA共通アドレス・スペースのベースにある作動構成レジスタに、該当するカード情報構造体(CIS)をロードすることによって、PCカード110がI/Oタイプの作動に合わせて構成される。I/Oアクセス・プロトコルは共通メモリ・アクセス・プロトコルと同様であるが、カード・イネーブル・ラインC0\_\_CE\*とC0\_\_WE\* (それぞれ、図11および図13では、C0\_\_RDCMD\*およびC0\_\_WRCMD\*で示す)の代わりに、カードI/OラインC0\_\_IORD\* (図11では、C0\_\_RDCMD\*で示す)とC0\_\_IOWR\* (図15参照)がそれぞれハンドシェイクに使用される点が異なっている。

【0064】 図15に戻ると、PCカード110はカード・ラインIOIS16\*を低にアサートし(図示せず)、カード110が16ビット幅の書き込みも実行できることを示す。これに応じて、アダプタ100はカード・イネーブル・ラインC0\_\_CE1\*を低にアサートして、カード・データ・ラインC0\_\_Dに出されたデータの両方のバイトがPCカード110に書き込まれるようにする。

【0065】 逆に、PCカード110がラインIOIS16\* (図示せず)を低にアサートしない場合、すなわち、PCカード110が16ビットI/O書き込みを行えない場合、カード・イネーブル・ラインC0\_\_CE1\*だけが低にアサートされるので、システム・バス・データ・ラインSB\_\_DATAからカード・データ・ラインC0\_\_Dに出される偶数バイトだけがPCカード110に書き込まれる。アダプタ100は次いで、上述のように、PCカード110に対して2つ以上のI/Oバイトの書き込みを実行することによって、ハーフワードI/O書き込みサイクルを完了する。この第2のバイトI/O書き込みアクセスが最初のバイトの書き込みと異なっているのは、カード・アドレス・ラインC0\_\_A[0]が「1」にセットされており、システム・バス・データ・ラインSB\_\_DATAからの奇数バイト値がカード・データ・ラインC0\_\_Dに出される点である。

【0066】 アダプタ100はWAIT機能を使用したPCカード110のI/O書き込みアクセスもサポートしている。図13に戻ると、WAIT機能を使用したI/O書き込みアクセスに対するイベントの順序はWAIT機能を使用した属性または共通メモリの書き込みアクセスのものと同様であるが、I/O書き込みアクセス中に、カード書き込みイネーブル・ラインC0\_\_WE\*ではなく、カードI/O書き込みイネーブル・ラインC0\_\_IOWR\* (図11では、C0\_\_WRCMD\*で示されている)が

20

アサートされる点が異なっている。PCカード110がカード待機ラインC0\_\_WAIT\*をアサートしたが、20マイクロ秒以内にアサート解除を行わなかった場合、I/O書き込みアクセスは終了し、カード110はホスト・コンピュータ10に対して状況変化割り込み(SCINT)を発生し、PCカード・アクセス・タイムアウト(PCTO)がインタフェース状況レジスタにセットされて、WAITタイムアウトが発生したことを示す。

【0067】 上述したように、PCMCIAソケット107をPCカード110の機能に応じて、属性/共通メモリ・モードまたはI/Oモードのいずれかで作動するように構成することができる。PCMCIAソケット107を属性/共通メモリ・モードに合わせて構成した場合、アダプタ100は次のイベントのいずれかを検出すると、状況変化割り込みを発生する。

【0068】 PCカード・アクセス・タイムアウト  
PCカード書き込み保護状況変化  
PCカード・レディービジー・状況変化  
PCカード・バッテリー状況変化  
PCカードの挿入または除去

【0069】 逆に、PCMCIAソケット107をI/Oモードに合わせて構成した場合、アダプタ100は次のイベントを検出すると、状況変化割り込みを発生する。

【0070】 PCカード・アクセス・タイムアウト  
PCカードの挿入または除去  
PCカードによる状況変化割り込みの発生

【0071】 アダプタ100が発生する割り込みは各種の機能をもたらす。たとえば、PCカード110をI/Oカードとして構成すると、PCカード110はカード状況の変化を検出したとき、カード状況変化ラインSTSC HG\* (図示せず)をアサートすることによって、ホスト・コンピュータ10に対して状況変化割り込みを発生する。アダプタ100はカード状況変化ラインSTSC HG\*を検出し、システム・バス55上でコンピュータ10に対して割り込みを発生する。ホスト・コンピュータ10は(I/Oタイプ)PCカード110のピン置換レジスタを読み取って、PCカード状況変化割り込み源を決定する。カード状況変化割り込み源の各々はアダプタ100によって個別にマスク可能であり、システム・バス・割り込み要求ラインSB\_\_INT[0]で利用可能である。その後、インタフェース状況レジスタ(0)の対応するステータス変化ビットに「1」を書き込むことによって、カード状況変化割り込みがコンピュータ10によってクリアされる。

【0072】 上述したように、本実施例はPCカードのホット・プラグングをサポートしているハードウェアも含んでいる。図8に戻ると、アダプタ100は制御信号を電源スイッチ105に与えて、2つの電源Vcc、VppのPCカード110に対する印加および除去を行う。このような構成によって、PCカード110のホッ

21

ト・プラグングが可能となる。すなわち、PCMCIAソケット107に関してPCカード110の有無を以下の態様で継続的に監視することによって、ホスト・コンピュータ10の電源を落とさずに、PCカード110をアダプタ100のソケットに対して安全に接続除去を行うことができる。

【0073】ホスト・コンピュータ10に電源を入れても、アダプタ100はPCMCIAソケット107に電力を供給しない。電源投入時にソケット107にPCカードが存在していることを検出すると、あるいは電源投入後にPCカード110が挿入されたことを検出すると、電力スイッチ回路105はスイッチ回路105の該当する電力用MOSFETをオンにすることによってPCカード110への電力の供給を開始する。逆に、アダプタ100に割込みが行われ、PCカード110の除去が通知された場合には、アダプタ100は該当する信号をCard\_\_Pwr\_\_Cntlラインを介して送り、この信号が電源スイッチ回路105に該当する電力用MOSFETをオフにさせ、これによってPCMCIAソケット107から電力を除去する。

【0074】実施例によっては、PCカード110は大容量記憶装置またはネットワーク装置であり、PROM106は格納されている、あるいはPCカード110がネットワーク接続によって検索するブート・イメージを使用してホスト・コンピュータ10をブートするように構成されていることもある。PROM106はタブルを識別するための独立したCISインタプリタも納めており、該タブルはブート中に装置識別および構成情報をPCカード、たとえば、PCカード110に与える。PROM常驻CISインタプリタは各PCカード、たとえば、PCカード110に対する少なくとも1つのデバイス情報ノードを備えたデバイス情報ツリーを構築する。さらに、PROM106はアダプタ100の機能およびシステム・リソースを定義する情報を納めている。オートブート・プロセスの詳細については、参照することによって全体が本明細書の一部を構成する、1992年2月25日出願の“METHOD AND APPARATUS FOR BOOTING A COMPUTER SYSTEM”という名称の米国特許願第07/842,007号を参照されたい。

【0075】他の実施例において、5ボルトから12ボルトへのDC-DCコンバータ107がある種のPCカードが作動に必要とする高電圧をもたらし、これによってアダプタ100の多用性が高められる。それ故、電力がカード110に与えられた後、アダプタ100はカード110の属性メモリ・スペースにあるカード情報構造体(CIS)を読み取り、PCカード110に関する情報を取得し、これによってPCカード110がアダプタ100のサポートできるタイプのものであることを確認する。さらに他の実施例においては、テスト・ポートがアダプタ100の内部診断を行う。

22

【0076】PCMCIA仕様バージョン2.1は直接メモリ・アクセス(DMA)タイプの作動をサポートしておらず、それ故、ホスト・コンピュータ10とPCカード110の間のデータ転送は本質的にプログラム式I/Oタイプ作動になる。しかしながら、PCMCIA仕様の今後のバージョンがDMAをサポートするようになった場合、当分野の技術者はアダプタ100にDMA機能を追加し、ソフトウェア・ドライバ34に適切な変更を施して、ホスト・コンピュータ10が初期設定後にホストCPU20に関わりなくホスト・メモリ30とPCカード110の間のデータ転送を開始できるようにすることができよう。

【0077】本発明を特定の実施例を使用して説明してきたが、本発明の精神および範囲から逸脱することなく、他の実施例、代替形、および改変形が当分野の技術者には明らかとなろう。たとえば、本発明のPCMCIAインタフェースをハードウェアとソフトウェアの様々な比率で実現することができる。それ故、上記の説明は単なる例であり、限定を目的としたものではない。本発明の真の範囲は特許請求の範囲で示されるものである。

#### 【図面の簡単な説明】

【図1】従来のSPARCホスト・コンピュータのハードウェア構成要素の図である。

【図2】ホスト・コンピュータのソフトウェア構成要素の図である。

【図3】PCMCIAインタフェースを有する従来技術のx86ベースのホスト・コンピュータの図である。

【図4】本発明の1実施例による、ホスト・コンピュータと結合されたPCMCIAインタフェースの図である。

【図5】図4のホスト・コンピュータで稼動するPCMCIAインタフェース・ソフトウェアの図である。

【図6】PCMCIAインタフェース・ソフトウェアの詳細を示すブロック図である。

【図7】ホスト・コンピュータで稼動しているユーザ・アプリケーションによるPCカードのアクセスを説明するための流れ図である。

【図8】本発明のPCMCIAハードウェア・アダプタの図である。

【図9】PCカードに割り振られたホスト・コンピュータのシステム・バス・アドレス・スペースを示すアドレス・マップである。

【図10】従来のPCMCIAの読取りおよび書き込みアクセス・サイクルを説明するための2つのタイミング図である。

【図11】属性メモリ・スペースおよび共通メモリ・スペースのPCカード読取りアクセスの図である。

【図12】WAIT機能による属性メモリ・スペースおよび共通メモリ・スペースのPCカード読取りアクセスの図である。

23

24

【図 1 3】 属性メモリ・スペースおよび共通メモリ・スペースの P C カード書き込みアクセスの図である。

【図 1 4】 W A I T 機能による属性メモリ・スペースおよび共通メモリ・スペースの P C カード書き込みアクセスの図である。

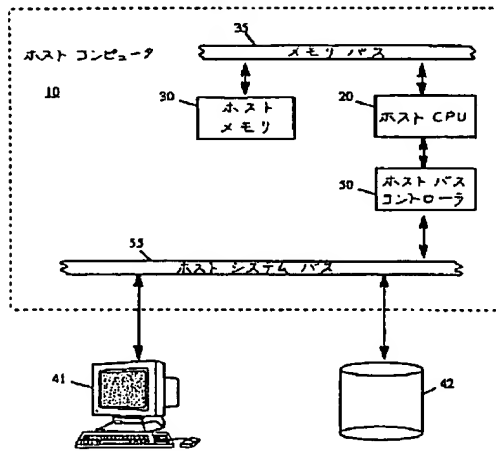
【図 1 5】 バイト・サイズ調整を行うが、W A I T 機能を使用しない I / O スペースの P C カード書き込みアクセスの図である。

【符号の説明】

10 ホスト・コンピュータ  
20 ホスト CPU  
30 ホスト・メモリ  
31 UNIX 互換ユーザ・アプリケーション  
32 UNIX カーネル  
33 デバイス・ドライバ  
34 P C M C I A インタフェース・ソフトウェア・ドライバ  
34 a ハードウェア非依存部分

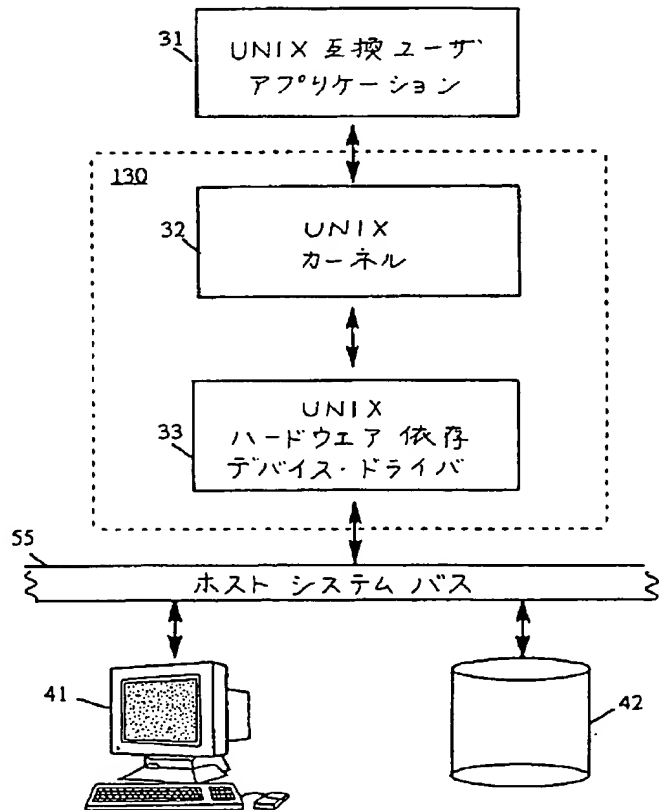
34 b ハードウェア依存部分  
55 システム・バス  
100 P C M C I A アダプタ  
100 a A S I C  
101 ホスト・バス・インタフェース・バッファ  
102 アダプタ・コア論理  
103 カード・インタフェース・バッファ  
105 電源スイッチ  
106 PROM  
107、108 P C M C I A ソケット  
110、120 P C カード  
135 UNIX オペレーティング・システム  
210 P C M C I A ネクサス・ドライバ  
220 カード・サービス層  
220 a カード情報構造体 ( C I S ) インタプリタ  
230 イベント・マネージャ  
250 P C M C I A アダプタ・ドライバ  
260、270 P C カード・ドライバ

【図 1】



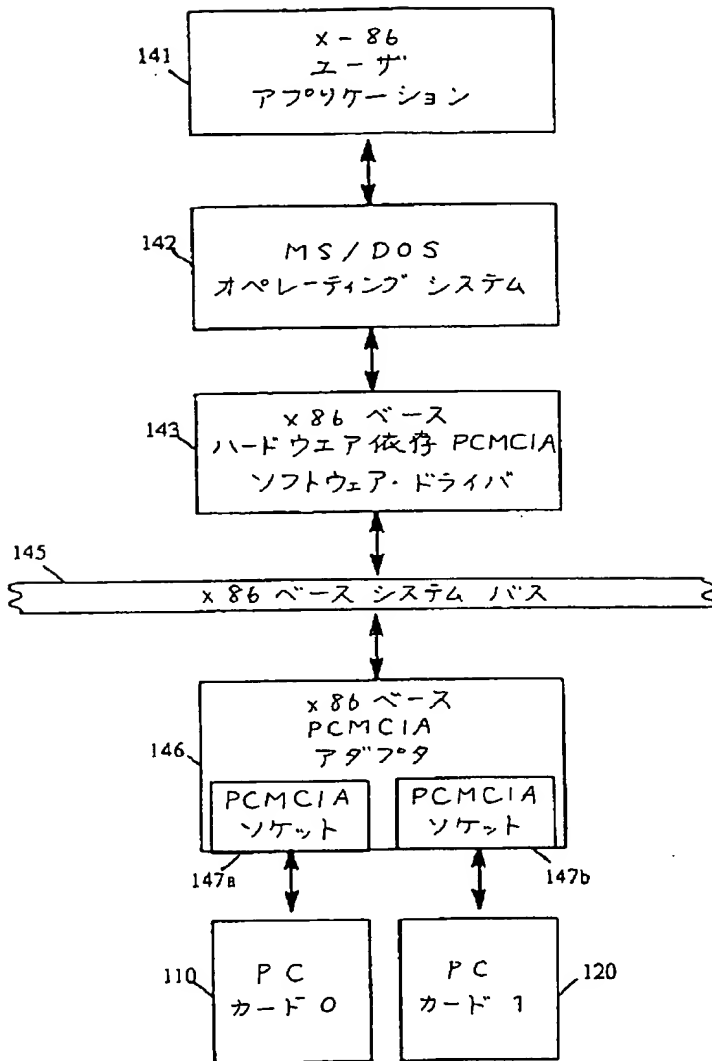
従来技術

【図 2】



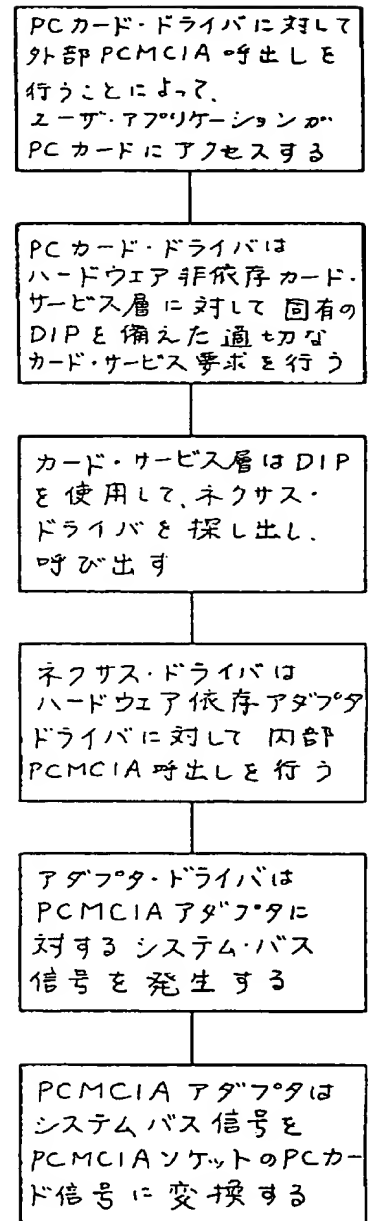
従来技術

【図 3】

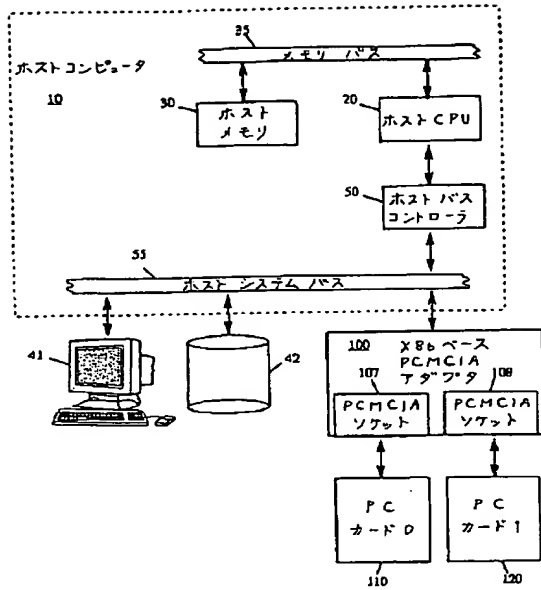


従来技術

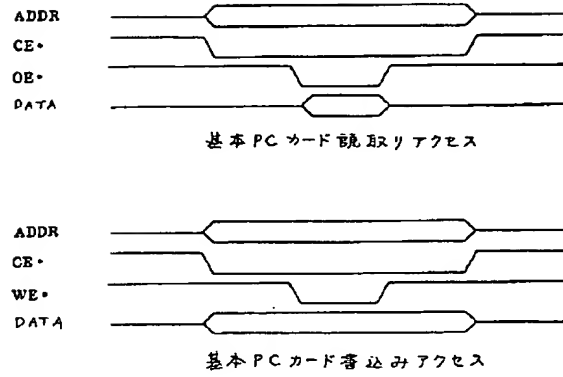
【図 7】



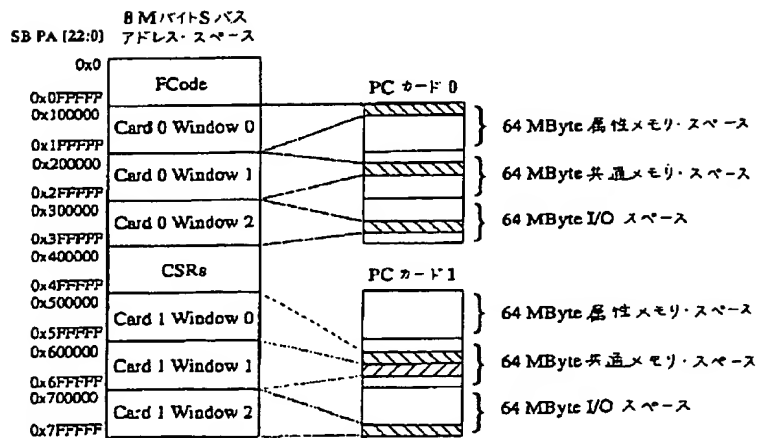
【図 4】



【図 10】

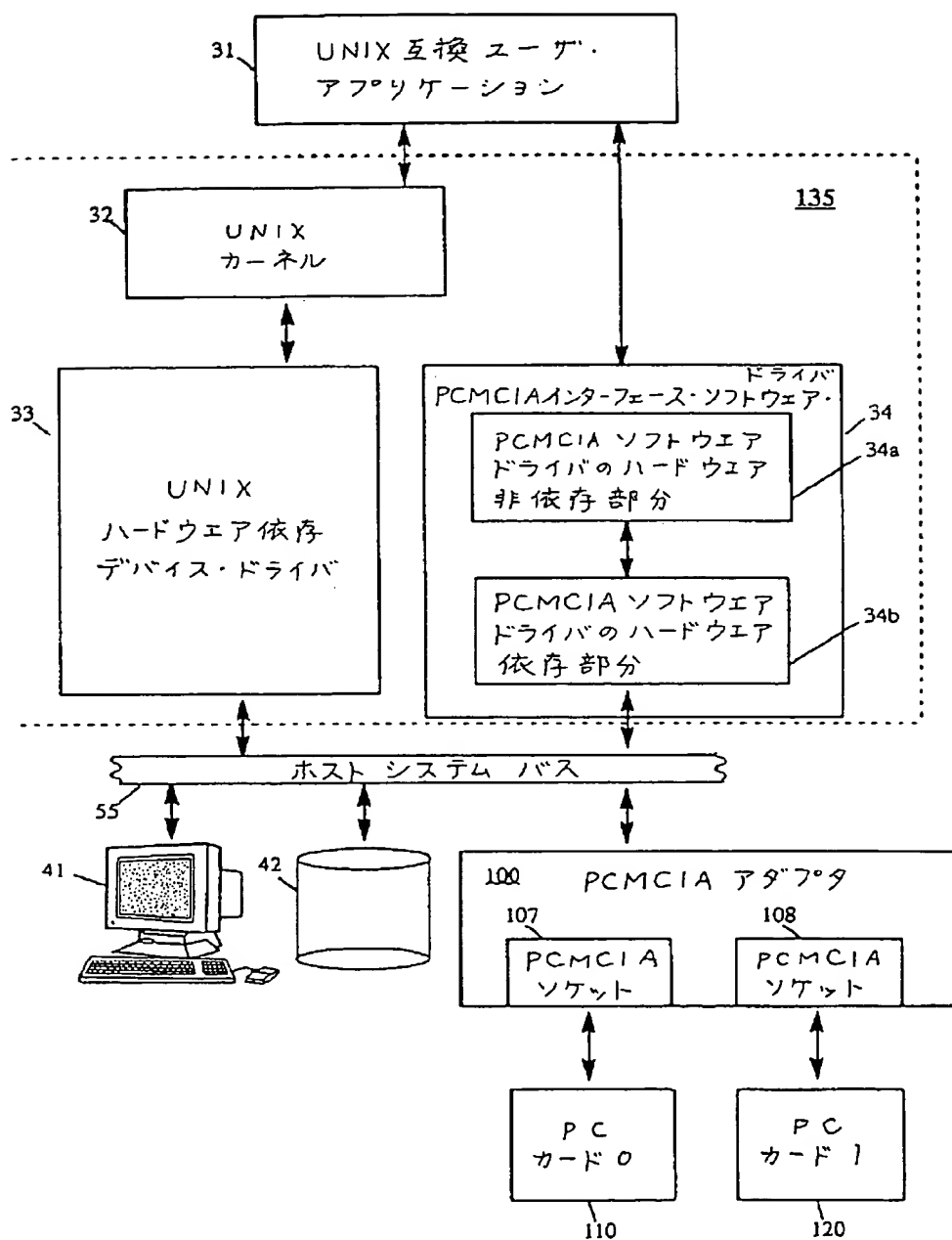


【図 9】

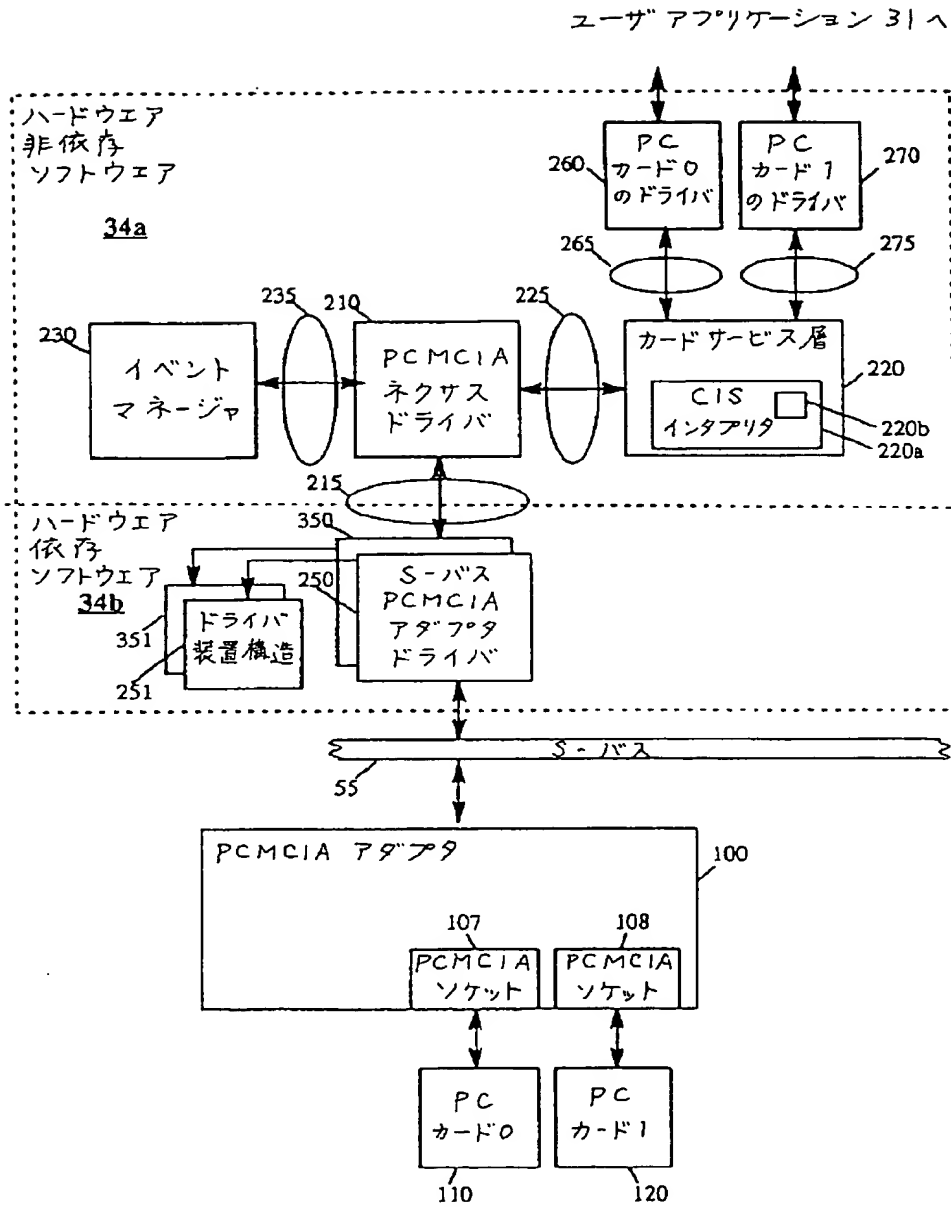




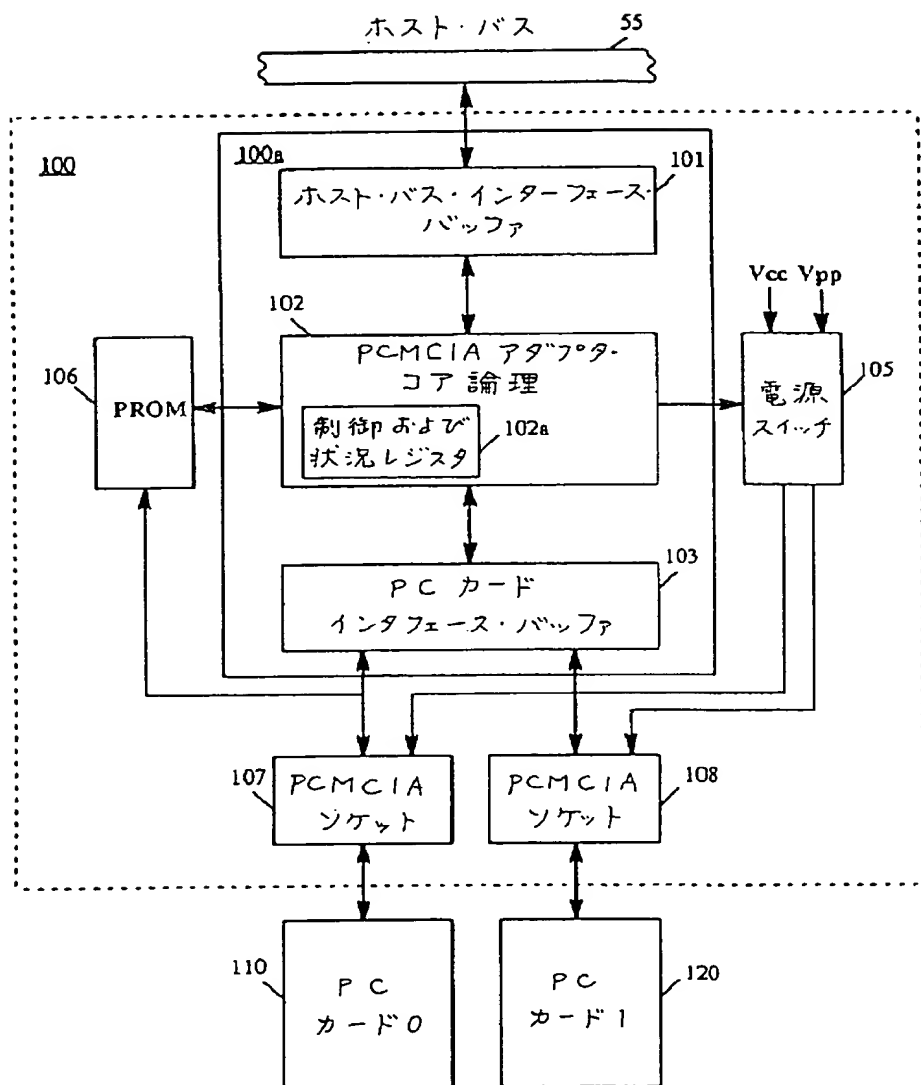
【図5】



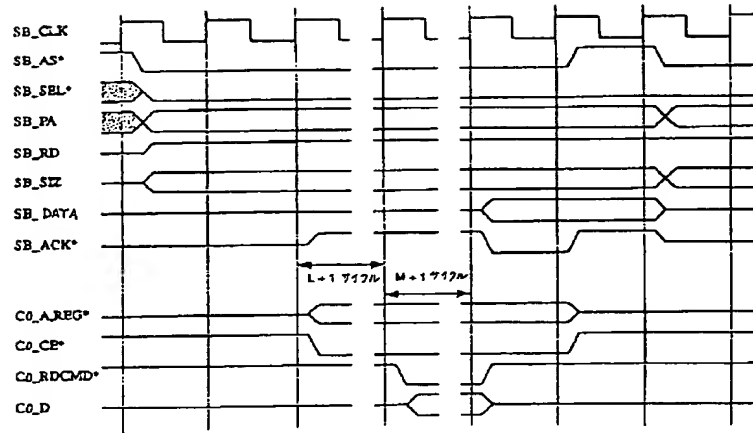
【図6】



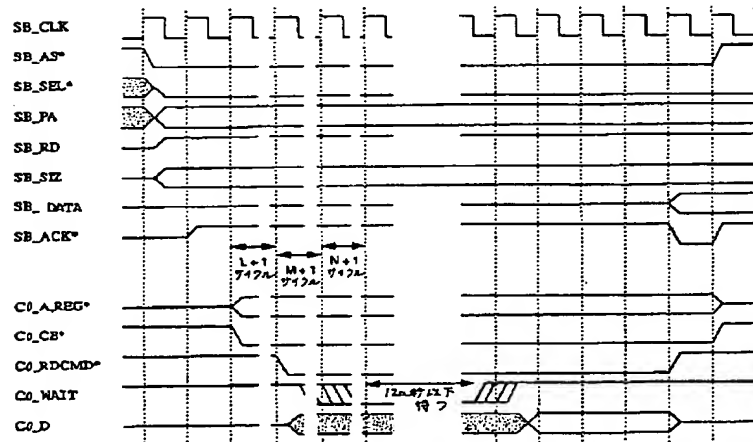
【図 8】



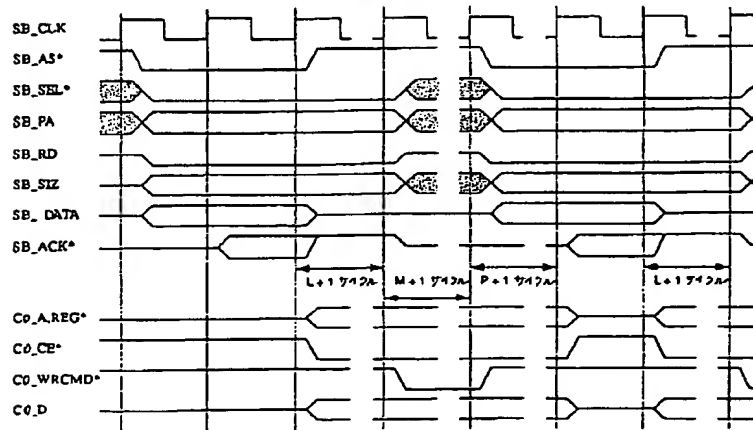
【図 1 1】



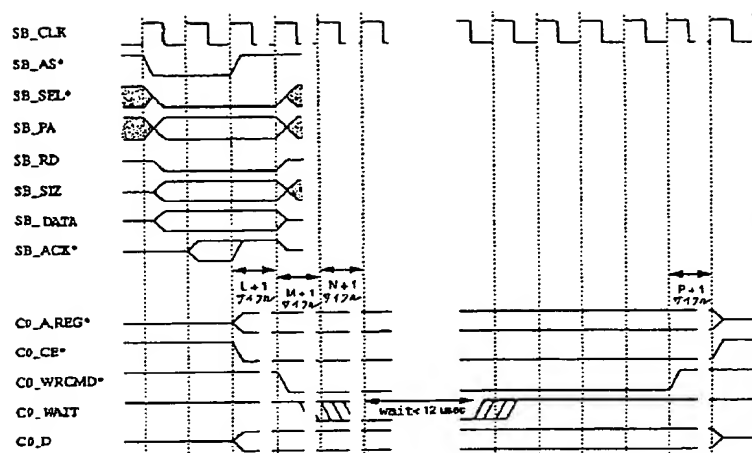
【図 1 2】



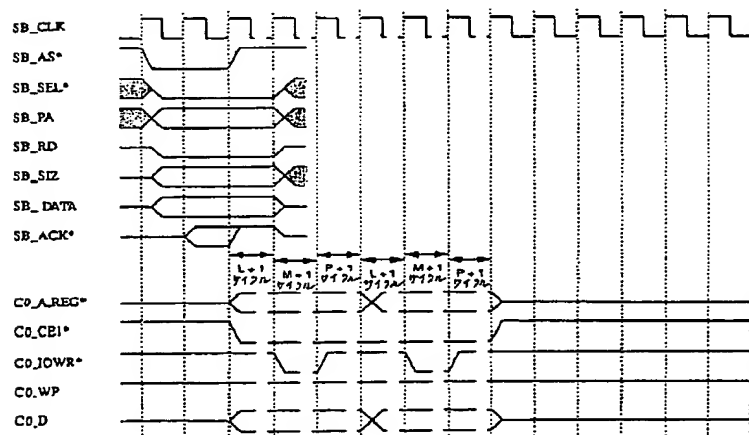
【図 1 3】



【図 14】



【図 15】



フロントページの続き

(72)発明者 ダグラス・マツカラム  
アメリカ合衆国 80027 コロラド州・ル  
イスヴィル・サウス ブキャナン アヴェ  
ニュー・197

(72)発明者 チャールズ・エフ・バットン, ジュニア  
アメリカ合衆国 94568 カリフォルニア  
州・ダブリン・ソリッド コート・11501  
(72)発明者 ドュオン・ミン・ヴォ  
アメリカ合衆国 95035 カリフォルニア  
州・ミルピタス・リッジモント ドライ  
ブ・1072